

# Analysis of Sum-Weight-like algorithms for averaging in Wireless Sensor Networks

Franck Iutzeler, Philippe Ciblat and Walid Hachem

## Abstract

Distributed estimation of the average value over a Wireless Sensor Network has recently received a lot of attention. Most papers consider single variable sensors and communications with feedback (e.g. peer-to-peer communications). However, in order to use efficiently the broadcast nature of the wireless channel, communications without feedback are advocated. To ensure the convergence in this feedback-free case, the recently-introduced Sum-Weight-like algorithms which rely on two variables at each sensor are a promising solution. In this paper, the convergence towards the consensus over the average of the initial values is analyzed in depth. Furthermore, it is shown that the squared error decreases exponentially with the time. In addition, a powerful algorithm relying on the Sum-Weight structure and taking into account the broadcast nature of the channel is proposed.

## I. INTRODUCTION

The recent years have seen a surge of signal processing and estimation technologies operating in stressful environments. These environments do not make possible the use of a fusion center so the units/sensors have to behave in a distributed fashion. In various applications, sensors have to communicate through wireless channels because of the lack of infrastructure. Hence, along with distributed computation, the problem of communicating between the different sensors to estimate a global value is a key issue and was pioneered by Tsitsiklis [2].

One of the most studied problems in Wireless Sensor Networks is the average computation of the initial measurements of the sensors. More precisely, each sensor wants to reach *consensus* over the mean of the initial values. A basic technique to address this problem, called *Random Gossip*, is to make the

The authors are with Institut Mines-Télécom/Télécom ParisTech ; CNRS LTCI. This work was partially granted by the French Defense Agency (DGA) and by the Télécom/Eurecom Carnot Institute. Some parts of the work introduced in this paper were presented at ICASSP 2012 [1].

sensors exchange their estimates in pairs and average them. This technique has been widely analyzed in terms of convergence and convergence speed in [3], [4], [5].

Finding more efficient exchange protocols has been a hot topic for the past few years; the proposed improvements were essentially twofold: i) exploiting the geometry of the network to have a more efficient mixing between the values (e.g. [6], [7], [8]) and ii) taking advantage of the broadcast nature of the wireless channels (e.g. [9] without feedback link, and [10] with feedback link). Whereas the use of network geometry has received a lot of attention, the use of the broadcast nature of the wireless channel is less studied albeit promising. Therefore, in our paper, we will focus on averaging algorithms taking into account the broadcast nature of the channel. In order to keep the number of communications as low as possible, we forbid the use of feedback links.

In the feedback-free context, one can mention [9]. However, even if the algorithm described in [9] converges quickly to a consensus, the reached value is incorrect. This can be explained by the fact that the sum of the sensor estimates is not constant over time. To overcome this problem, Franceschelli *et al.* [11] proposed to use well-chosen updates on two local variables per sensor while using the broadcast nature of the channel without feedback link. A more promising alternative is to use the *Sum-Weight* scheme proposed by Kempe [12] and studied more generally by Bénézit [13]. In this setup, two local variables are also used: one representing the *sum* of the received values and the other representing the *weight* of the sensor (namely, the proportion of the sensor activity compared to the others). The two variables are transmitted at each iteration and both are updated in the same manner. The wanted estimate is then the quotient of these values. The convergence of this class of algorithms (without necessarily sum-conservation) has been proven in [12], [13]. In contrast, their convergence speed has never been theoretically evaluated except in [12] for a very specific case.

The goal of this paper is to theoretically analyze the convergence speed for any Sum-Weight-like algorithm. As a by-product, we obtain necessary and sufficient condition for the convergence. In addition, we propose a new Sum-Weight-like algorithm based on broadcasting which outperforms existing ones.

This paper is organized as follows: the notations and assumptions on the network model and on the Sum-Weight-like algorithms are provided in Section II. Section III is dedicated to the theoretical analysis of the squared error of the algorithms and provides the main contributions of the paper. In Section IV, we propose new Sum-Weight-like algorithms. In Section V, we compare our results with previous derivations done in the literature for the Sum-Weight-like algorithms as well as the algorithms based on the exchange of a single variable between the nodes. Section VI is devoted to numerical illustrations. Concluding remarks are drawn in Section VII.

## II. MODEL AND ASSUMPTIONS

### A. Network model

The sensor network will be modeled by a directed graph  $\mathcal{G} = (V, E)$ ,  $V$  being the set of vertices/sensors and  $E$  being the set of edges which models the possible links between the sensors. We also define the adjacency matrix  $\mathbf{A}$  of  $\mathcal{G}$  as the  $N \times N$  matrix such that  $(\mathbf{A})_{ij}$  equals 1 if there is an edge from  $i$  to  $j$  and 0 otherwise. We define the neighborhood of each sensor  $i$  as follows  $\mathcal{N}_i = \{j \in V | (i, j) \in E\}$ . Let  $d_i = |\mathcal{N}_i|$  denote the degree of the sensor  $i$  where  $|\mathcal{A}|$  represents the cardinality of the set  $\mathcal{A}$ . Let  $d_{max} = \max_i d_i$  be the maximum degree. Let  $\mathbf{D} = \text{diag}(d_1, \dots, d_N)$  and  $\mathbf{L} = \mathbf{D} - \mathbf{A}$  be the degree matrix and the Laplacian matrix respectively [14].

Every sensor  $i$  has an initial value  $x_i(0)$  and we define  $\mathbf{x}(0) = [x_1(0), \dots, x_N(0)]^T$  where the superscript  $T$  stands for the transposition. The goal of the network is to communicate through the edges of the underlying graph to reach consensus over the mean of the initial values of the sensors. A communication and estimation step will be referred to as an *update*.

We will assume that the network follows a discrete time model such that the time  $t$  is the time of the  $t$ -th update. As an example, every sensor could be activated by an independent Poisson clock. The time would then be counted as the total number of clock ticks across the network. We will denote  $x_i(t)$  the  $i$ -th sensor estimate at time  $t$  and  $\mathbf{x}(t) = [x_1(t), \dots, x_N(t)]^T$ .

### B. Averaging Algorithms

The goal of averaging algorithms is to make the vector of estimates  $\mathbf{x}(t)$  converge to  $x_{ave}\mathbf{1}$ , also known as the *consensus vector*, where  $\mathbf{1}$  is the length- $N$  vector of ones and  $x_{ave} = (1/N)\mathbf{1}^T\mathbf{x}(0)$  is the average of the initial values of the sensors. In the present state-of-the-art, two classes of algorithms exist and are described below.

1) *Class of Random Gossip algorithms*: In standard gossip algorithms (e.g. [4]), sensors update their estimate according to the equation  $\mathbf{x}(t+1)^T = \mathbf{x}(t)^T \mathbf{K}(t)$  where the  $\mathbf{K}(t)$  are doubly-stochastic matrices. We recall that a matrix  $\mathbf{K}$  is said row-stochastic (resp. column-stochastic) when all its elements are non-negative and when  $\mathbf{K}\mathbf{1} = \mathbf{1}$  (resp.  $\mathbf{K}^T\mathbf{1} = \mathbf{1}$ ). A matrix which is both row- and column-stochastic is said to be doubly-stochastic. Since two sensors can exchange information only across the edges of the graph, for any  $i \neq j$ ,  $(\mathbf{K}(t))_{ij}$  cannot be positive if  $(\mathbf{A})_{ij} = 0$ . From an algorithmic point of view, the row-stochasticity implies that the sum of the values is unchanged:  $\mathbf{x}(t+1)^T\mathbf{1} = \mathbf{x}(t)^T\mathbf{K}(t)\mathbf{1} = \mathbf{x}(t)^T\mathbf{1}$  whereas the column-stochasticity implies that the consensus is stable: if  $\mathbf{x}(t) = c\mathbf{1}$ , then  $\mathbf{x}(t+1)^T = \mathbf{x}(t)^T\mathbf{K}(t) =$

$c\mathbf{1}^T\mathbf{K}(t) = c\mathbf{1}^T$ . For these reasons, double-stochasticity is desirable. However using doubly-stochastic matrices implies a feedback which is not always possible. In particular if a sensor sends information to multiple neighbors, the feedback message might raise multiple access problems. Similarly, if the message is sent through a long route within the network, the same route may not exist anymore for feedback in the context of mobile wireless networks. As these algorithms only rely on the exchanges of one variable per sensor, they will be called *single-variate* algorithms in the rest of the paper.

2) *Class of Sum-Weight algorithms*: To overcome this drawback, a possible method is to use two variables : one representing the *sum* of the received values and another representing the relative *weight* of the sensor. For the sensor  $i$  at time  $t$ , they be respectively written  $s_i(t)$  and  $w_i(t)$ . Writing  $\mathbf{s}(t) = [s_1(t), \dots, s_N(t)]^T$  and  $\mathbf{w}(t) = [w_1(t), \dots, w_N(t)]^T$ , both variables will be modified by the same update matrix,  $\mathbf{s}(t+1)^T = \mathbf{s}(t)^T\mathbf{K}(t)$  and  $\mathbf{w}(t+1)^T = \mathbf{w}(t)^T\mathbf{K}(t)$ . Finally, the estimate of sensor  $i$  at time  $t$  will be the quotient of the two variables,  $x_i(t) \triangleq s_i(t)/w_i(t)$ . The initialization is done as follows:

$$\begin{cases} \mathbf{s}(0) = \mathbf{x}(0) \\ \mathbf{w}(0) = \mathbf{1}. \end{cases} \quad (1)$$

For the sake of convergence we will need an important property: *Mass Conservation*

$$\begin{cases} \sum_{i=1}^N s_i(t) = \sum_{i=1}^N x_i(0) = Nx_{ave} \\ \sum_{i=1}^N w_i(t) = N. \end{cases} \quad (2)$$

This clearly rewrites as  $\forall t > 0, \mathbf{K}(t)\mathbf{1} = \mathbf{1}$  which corresponds to sum-conservation as in classic gossip algorithms and leads to row-stochastic updates matrices.

### C. Notations for the Sum-Weight scheme

Let us now introduce some useful notations along with some fundamental assumptions for convergence in the Sum-Weight scheme. Given two vectors  $\mathbf{a}$  and  $\mathbf{b}$  with the same size, we denote by  $\mathbf{a}/\mathbf{b}$  the vector of the elementwise quotients. The Sum-Weight algorithm is described by the following equations:

$$\begin{aligned} \mathbf{x}(t) &\triangleq \frac{\mathbf{s}(t)}{\mathbf{w}(t)} = \left[ \frac{s_1(t)}{w_1(t)}, \dots, \frac{s_N(t)}{w_N(t)} \right]^T \\ \begin{cases} \mathbf{s}^T(t+1) = \mathbf{s}^T(t)\mathbf{K}(t) = \mathbf{x}^T(0)\mathbf{P}(t) \\ \mathbf{w}^T(t+1) = \mathbf{w}^T(t)\mathbf{K}(t) = \mathbf{1}^T\mathbf{P}(t) \end{cases} \end{aligned}$$

with  $\mathbf{P}(t) = \mathbf{K}(1)\mathbf{K}(2) \dots \mathbf{K}(t)$ .

In the following, the matrix inequalities will be taken elementwise so that  $\mathbf{M} > 0$  (resp.  $\mathbf{M} \geq 0$ ) means that the matrix  $\mathbf{M}$  is (elementwise) *positive* (resp. *non-negative*). We recall that a non-negative

matrix  $\mathbf{M}$  is said to be *primitive* if  $\mathbf{M}^m > 0$  for some  $m \geq 1$  (see [15, Chap 8.5] for details). We will denote the Kronecker product by ' $\otimes$ '.

We can notice that reaching consensus is equivalent for  $\mathbf{x}(t)$  to converge to the consensus line  $c\mathbf{1}$  where  $c$  is consensus value. For this reason, it is useful to define  $\mathbf{J} = (1/N)\mathbf{1}\mathbf{1}^T$  the orthogonal projection matrix to the subspace spanned by  $\mathbf{1}$  and  $(\mathbf{I} - \mathbf{J})$  the orthogonal projection matrix to the complementary subspace which can be seen as the error hyperplane. The matrix  $\mathbf{I}$  is the identity matrix with appropriate size.

In order to intuitively understand the algorithm behavior, let us decompose  $\mathbf{x}^T(t)$  as follows

$$\begin{aligned}
 \mathbf{x}^T(t) &= \frac{\mathbf{s}^T(t)}{\mathbf{w}^T(t)} = \frac{\mathbf{x}^T(0)\mathbf{P}(t)}{\mathbf{w}^T(t)} \\
 &= \frac{\mathbf{x}^T(0)\mathbf{J}\mathbf{P}(t)}{\mathbf{w}^T(t)} + \frac{\mathbf{x}^T(0)(\mathbf{I} - \mathbf{J})\mathbf{P}(t)}{\mathbf{w}^T(t)} \\
 &= \frac{x_{ave}\mathbf{1}^T\mathbf{P}(t)}{\mathbf{1}^T\mathbf{P}(t)} + \frac{\mathbf{x}^T(0)(\mathbf{I} - \mathbf{J})\mathbf{P}(t)}{\mathbf{w}^T(t)} \\
 &= x_{ave}\mathbf{1}^T + \frac{\mathbf{x}^T(0)(\mathbf{I} - \mathbf{J})\mathbf{P}(t)}{\mathbf{w}^T(t)} \tag{3}
 \end{aligned}$$

Obviously, the algorithm will converge to the right consensus if the second term in the right hand side vanishes. Actually, under some mild assumptions related to the connectedness of the network, we expect the numerator which corresponds to a projection on the error hyperplane will converge to zero at an exponential rate while all the elements of  $\mathbf{w}(t)$  are of order one. Proving these results will be the core of the paper.

#### D. Assumptions on the update matrices $\mathbf{K}(t)$

First, we will always assume that both following conditions will be satisfied by any update matrix associated with a Sum-Weight like algorithm.

- (A1) Matrices  $(\mathbf{K}(t))_{t>0}$  are independent and identically distributed (i.i.d.), and row-stochastic. The matrix  $\mathbf{K}(t)$  is valued in a set  $\mathcal{K} = \{\mathbf{K}_i\}_{i=1..M}$  of size  $M < \infty$ . Also,  $p_i = \mathbb{P}[\mathbf{K}(t) = \mathbf{K}_i] > 0$ .
- (A2) Any matrix in  $\mathcal{K}$  has a strictly positive diagonal.

The first assumption is just a reformulation of the *mass conservation* property introduced in section II-B2 along with the assumption of a finite number of actions across the network. This assumption is reasonable when one assumes that each sensor performs a finite number of actions. The second assumption forces

every sensor to keep a part of the information it had previously. We also define

$$\begin{cases} m_{\mathcal{K}} = \min_{i,j,k} \left\{ (\mathbf{K}_k)_{ij} : (\mathbf{K}_k)_{ij} > 0 \right\}, \\ p_{\mathcal{K}} = \min_k \{ \mathbb{P}[\mathbf{K}(t) = \mathbf{K}_k] \} = \min_k p_k > 0. \end{cases} \quad (4)$$

In addition to both previous assumptions, we will see that next assumption plays a central role in the convergence analysis of any Sum-Weight like algorithm.

**(B)**  $\mathbb{E}[\mathbf{K}] = \sum_{i=1}^M p_i \mathbf{K}_i$  is a primitive matrix.

In terms of graph theory, matrix  $\mathbb{E}[\mathbf{K}]$  represents a weighted directed graph (see [15, Def. 6.2.11]). Since it is primitive, this graph is strongly connected (see [15, Cor. 6.2.18] and [20]). Observe that this graph contains a self-loop at every node due to Assumption **(A2)**. In fact, the matrix  $\mathbf{A} + \mathbf{I}$  coincides with the so-called *indicator matrix* ([15, Def. 6.2.10]) of  $\mathbb{E}[\mathbf{K}]$ .

### III. MATHEMATICAL RESULTS

#### A. Preliminary results

The assumption **(B)** can be re-written in different ways thanks to the next Lemma.

**Lemma 1.** *Under assumptions **(A1)** and **(A2)**, the following propositions are equivalent to **(B)** :*

**(B1)**  $\forall (i, j) \in \{1, \dots, N\}^2, \exists L_{ij} < N$  and a realization of  $\mathbf{P}(L_{ij})$  verifying  $\mathbf{P}(L_{ij})_{i,j} > 0$ .

**(B2)**  $\exists L < 2N^2$  and a realization of  $\mathbf{P}(L)$  which is a positive matrix.

**(B3)**  $\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] = \sum_{i=1}^M p_i \mathbf{K}_i \otimes \mathbf{K}_i$  is a primitive matrix.

The proof is reported in Appendix A. This Lemma will be very useful in the sequel since it enables us to interpret the Assumption **(B)** in various manners.

Our approach for analyzing the convergence of Sum-Weight algorithms is inspired by [12] (with a number of important differences explained below) and so relies on the analysis of the Squared Error (SE). Actually, the Squared Error can be upper-bounded by a product of two terms as follows

$$\|\mathbf{x}(t) - x_{ave} \mathbf{1}\|_2^2 = \sum_{i=1}^N |x_i(t) - x_{ave}|^2 = \sum_{i=1}^N \frac{1}{w_i(t)^2} |s_i(t) - x_{ave} w_i(t)|^2 \quad (5)$$

$$\begin{aligned} &= \sum_{i=1}^N \frac{1}{w_i(t)^2} \left| \sum_{j=1}^N x_j(0) \mathbf{P}_{ji}(t) - \frac{1}{N} \sum_{k=1}^N x_k(0) \cdot \sum_{l=1}^N \mathbf{P}_{li}(t) \right|^2 \\ &\leq \Psi_1(t) \Psi_2(t) \end{aligned} \quad (6)$$

$$\text{with} \quad \Psi_1(t) = \frac{\|\mathbf{x}(0)\|_2^2}{[\min_k w_k(t)]^2} \quad (7)$$

$$\Psi_2(t) = \sum_{i=1}^N \sum_{j=1}^N \left| (\mathbf{P}^T(t) (\mathbf{I} - \mathbf{J}))_{ij} \right|^2. \quad (8)$$

Notice that the decomposition done in Eq. (6) mimics Eq. (3) for the Squared Error.

From now, our main contributions will be to understand the behavior of both terms  $\Psi_1(t)$  and  $\Psi_2(t)$  when  $t$  is large. In Section III-B, we will prove that  $\Psi_1(t)$  can be upper bounded infinitely often. The term  $\Psi_2(t)$  represents the projection of the current sensor values on the orthogonal space to the consensus line. The analysis of this term is drawn in Section III-C.

### B. Analysis of $\Psi_1(t)$

This term depends on the inverse of the minimum of the sensors weights (see Eq. (7)) and thus can increase quickly. However, the sensors frequently exchange information and hence spread their weight so the probability that a node weight keeps decreasing for a long time is very small. We will work on this probability and show that it can be made as small as one wants considering a sufficiently long amount of time. This will enable us to prove that  $\Psi_1(t)$  will be infinitely often lower than a finite constant. To obtain these results, some preliminary lemmas are needed.

First, we will focus on the behavior of the nodes weights and especially on their minimum. One can remark that at every time  $t$  there is at least one node whose weight is greater than or equal to 1 (as the weights are non-negative and  $\forall t > 0, \sum_i w_i(t) = N$  because of the mass conservation exhibited in Eq. (2)). As  $\mathbf{w}(t_0 + t)^T = \mathbf{w}(t_0)^T \mathbf{P}(t_0, t_0 + t)$  where  $\mathbf{P}(t_0, t_0 + t) \triangleq \mathbf{K}(t_0) \dots \mathbf{K}(t_0 + t)$ , it is interesting to focus on i) the minimum non-null value of  $\mathbf{P}(t_0, t_0 + t)$  and ii) on the instants where  $\mathbf{P}(t_0, t_0 + t)$  is positive.

**Lemma 2.** *For all  $t, t_0 > 0$ , all the non-null coefficients of  $\mathbf{P}(t_0, t_0 + t)$  are greater than or equal to  $(m_{\mathcal{K}})^t$ .*

*Proof:* Let us recall that  $m_{\mathcal{K}}$  is the smallest non-null entry of all the matrices belonging to the set  $\mathcal{K}$  as defined in Eq. (4). Let us consider the random matrix  $\mathbf{P}(t)$  (as the matrix choice is i.i.d., we drop the offset  $t_0$ ). We will then prove this result by induction. It is trivial to see that every non-null coefficient

of  $\mathbf{P}(1) = \mathbf{K}(1)$  is greater than  $m_{\mathcal{K}}$  and as

$$(\mathbf{P}(t))_{i,j} = \sum_{k=1}^N (\mathbf{P}(t-1))_{i,k} (\mathbf{K}(t))_{k,j},$$

it is obvious that if  $(\mathbf{P}(t))_{i,j} > 0$ , then there is a term in the sum that is positive (we remind that all the coefficient here are non-negative). This term is the product of a positive coefficient of  $\mathbf{P}(t-1)$  and a positive coefficient of  $\mathbf{K}(t)$ . Hence, if all the non-null coefficients of  $\mathbf{P}(t-1)$  are greater than  $(m_{\mathcal{K}})^{t-1}$ , then any non-null coefficient of  $\mathbf{P}(t)$  is greater than  $(m_{\mathcal{K}})^{t-1} \cdot m_{\mathcal{K}} = (m_{\mathcal{K}})^t$ . So, by induction, we have that  $\forall t > 0$  every non-null coefficient of  $\mathbf{P}(t)$  is greater than  $(m_{\mathcal{K}})^t$ . ■

Thanks to Item **(B2)** of Lemma 1, there is a finite  $L$  such that there exists a realization of  $\mathbf{P}(L)$  which is a positive matrix. Considering the time at multiples of  $L$ , we know that for any  $n$ , if  $\mathbf{P}(nL+1, (n+1)L) > 0$  then for all  $i$ ,  $w_i((n+1)L) \geq m_{\mathcal{K}}^L$ . Let us define the following stopping times:

$$\begin{cases} \tau_0 = 0 \\ \tau_n = L \times \min \left\{ j : \sum_{k=1}^j \mathbb{1}_{\{\mathbf{P}(kL+1, (k+1)L) > 0\}} = n \right\} \end{cases}$$

where  $\mathbb{1}_E$  is the indicator function of event  $E$ . And,

$$\Delta_n = \tau_n - \tau_{n-1} \quad n = 1, \dots, \infty.$$

The  $\mathbb{1}_{\{\mathbf{P}(kL+1, (k+1)L) > 0\}}$  are i.i.d. Bernoulli random variables with strictly positive parameter  $p$ . Thus the inter-arrival times  $\Delta_n$  are i.i.d. and geometrically distributed *i.e.*  $\mathbb{P}[\Delta_1 = k] = p^{k-1}(1-p)$  for  $k \geq 1$ . Observe that the  $(\tau_n)_{n>0}$  are all finite and converge to infinity with probability one. We then have proven the following result:

**Proposition 1.** *Under Assumptions (A1), (A2), and (B), there exists a sequence of positive i.i.d. geometrically distributed random variables  $(\Delta_n)_{n>0}$  such that for all  $n > 0$*

$$\Psi_1(\tau_n) \leq \|\mathbf{x}(0)\|_2^2 (m_{\mathcal{K}})^{-2L}$$

where  $\tau_n = \sum_{k=1}^n \Delta_k$ .

### C. Analysis of $\Psi_2(t)$

This section deals with new results about  $\Psi_2(t)$ . These results extend dramatically those given in [12] since we consider more general models for  $\mathbf{K}(t)$  and any type of connected graph. According to Eq. (8), we have

$$\Psi_2(t) = \|(\mathbf{I} - \mathbf{J}) \mathbf{P}(t)\|_F^2 \quad (9)$$



where  $\|\bullet\|_F$  denotes the Frobenius matrix norm.

One technique (used in *e.g.* [4]) consists in writing  $\mathbb{E}[\Psi_2(t)] = \text{Trace}((\mathbf{I} - \mathbf{J})\mathbb{E}[\mathbf{P}(t)\mathbf{P}^T(t)](\mathbf{I} - \mathbf{J}))$  thanks to Eq. (9) and finding a linear recursion between  $\mathbb{E}[\Psi_2(t)|\Psi_2(t-1)]$  and  $\Psi_2(t-1)$ . However this technique does not work in the most general case <sup>1</sup>.

Therefore, as proposed alternatively in [4] (though not essential in [4]) in the context of Random-Gossip Algorithms (see Section II-B1), we write  $\Psi_2(t)$  with respect to a more complicated matrix for which the recursion property is easier to analyze. Indeed, recalling that for any matrix  $\mathbf{M}$ ,

$$\begin{aligned}\|\mathbf{M}\|_F^2 &= \text{Trace}(\mathbf{M}\mathbf{M}^T) \\ \text{and } \text{Trace}(\mathbf{M} \otimes \mathbf{M}) &= (\text{Trace}(\mathbf{M}))^2\end{aligned}$$

one can find that

$$\Psi_2(t) = \|\Xi(t)\|_F$$

with

$$\Xi(t) = (\mathbf{I} - \mathbf{J})\mathbf{P}(t) \otimes (\mathbf{I} - \mathbf{J})\mathbf{P}(t).$$

By remarking that  $(\mathbf{I} - \mathbf{J})\mathbf{P}(t)(\mathbf{I} - \mathbf{J}) = (\mathbf{I} - \mathbf{J})\mathbf{P}(t)$ , and by using standard properties on the Kronecker product, we have

$$\begin{aligned}\Xi(t) &= (\mathbf{I} - \mathbf{J})\mathbf{P}(t-1)(\mathbf{I} - \mathbf{J})\mathbf{K}(t) \otimes (\mathbf{I} - \mathbf{J})\mathbf{P}(t-1)(\mathbf{I} - \mathbf{J})\mathbf{K}(t) \\ &= \Xi(t-1)[((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}))(\mathbf{K}(t) \otimes \mathbf{K}(t))].\end{aligned}\tag{10}$$

By considering the mathematical expectation given the natural filtration of the past events  $\mathcal{F}_{t-1} = \sigma(\mathbf{K}(1), \dots, \mathbf{K}(t-1))$ , we obtain

$$\mathbb{E}[\Xi(t)|\mathcal{F}_{t-1}] = \Xi(t-1)((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}))\mathbb{E}[\mathbf{K} \otimes \mathbf{K}].$$

As  $\Xi(0) = (\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})$  and  $((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}))^2 = (\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})$ , we finally have

$$\mathbb{E}[\Xi(t)] = \mathbf{R}^t.\tag{11}$$

<sup>1</sup>We have  $\mathbb{E}[\Psi_2(t)|\Psi_2(t-1)] = \text{Trace}((\mathbf{I} - \mathbf{J})\mathbf{P}(t-1)(\mathbf{I} - \mathbf{J})\mathbb{E}[\mathbf{K}\mathbf{K}^T](\mathbf{I} - \mathbf{J})\mathbf{P}(t-1)(\mathbf{I} - \mathbf{J}))$ . By introducing the matrix  $\mathbf{M} = (\mathbf{I} - \mathbf{J})\mathbb{E}[\mathbf{K}\mathbf{K}^T](\mathbf{I} - \mathbf{J})$ , it is easy to link  $\mathbb{E}[\Psi_2(t)|\Psi_2(t-1)]$  with  $\Psi_2(t-1)$  since  $\mathbb{E}[\Psi_2(t)|\Psi_2(t-1)] \leq \|\mathbf{M}\|_{sp}\Psi_2(t-1)$  where  $\|\bullet\|_{sp}$  is the spectral norm (see [15, Chap. 7.7] for details). Unfortunately, in some cases,  $\|\mathbf{M}\|_{sp}$  can be greater than 1; indeed for the *BWGossip* algorithm (introduced in Section IV-A), one can have  $\|\mathbf{M}\|_{sp} > 1$  for some underlying graphs. Nevertheless, this *BWGossip* algorithm converges as we will see later. As a consequence, the inequality  $\mathbb{E}[\Psi_2(t)|\Psi_2(t-1)] \leq \|\mathbf{M}\|_{sp}\Psi_2(t)$  is not tight enough to prove a general convergence result and another way has to be found.

with

$$\mathbf{R} = ((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})) \cdot \mathbb{E}[\mathbf{K} \otimes \mathbf{K}]. \quad (12)$$

Now one can find a simple relationship between  $\mathbb{E}[\Psi_2(t)]$  and the entries of the matrix  $\mathbb{E}[\Xi(t)]$  by considering  $\mathbf{Q}(t) = (\mathbf{I} - \mathbf{J}) \mathbf{P}(t)$  and  $(\mathbf{Q}(t))_{i,j} = q_{ij}(t)$ . After simple algebraic manipulations, we show that

$$(\mathbb{E}[\Xi(t)])_{i+(k-1)N, j+(l-1)N} = \mathbb{E}[q_{ij}(t)q_{kl}(t)], \quad \forall (i, j, k, l) \in \{1, \dots, N\}^4.$$

According to Eq. (9), we have  $\mathbb{E}[\Psi_2(t)] = \mathbb{E}[\|\mathbf{Q}(t)\|_F^2]$  which implies that

$$\mathbb{E}[\Psi_2(t)] = \sum_{i,j=1}^N \mathbb{E}[q_{ij}^2(t)] = \sum_{i,j=1}^N (\mathbb{E}[\Xi(t)])_{i+(i-1)N, j+(j-1)N}.$$

As a consequence, the behavior of the entries of  $\mathbb{E}[\Xi(t)]$  drives the behavior of  $\mathbb{E}[\Psi_2(t)]$ .

Let us define the  $L_\infty$  vector norm on  $N \times N$  matrices as  $\|\mathbf{M}\|_\infty = N \max_{1 \leq i, j \leq N} |m_{ij}|$ . The norm  $\|\bullet\|_\infty$  is a matrix norm (see [15, Chap. 5.6]) and hence is submultiplicative. Now, using the Jordan normal form of  $\mathbf{R}$  (see [15, Chap. 3.1 and 3.2]), we get that there is an invertible matrix  $\mathbf{S}$  such that

$$\|\mathbf{R}^t\|_\infty = \|\mathbf{S} \mathbf{\Lambda}^t \mathbf{S}^{-1}\|_\infty \leq \|\mathbf{S}\|_\infty \|\mathbf{S}^{-1}\|_\infty \|\mathbf{\Lambda}^t\|_\infty \quad (13)$$

where  $\mathbf{\Lambda}$  is the Jordan matrix associated with  $\mathbf{R}$ .

After some computations, it is easy to see that the absolute value of all the entries of  $\mathbf{\Lambda}^t$  are bounded in the following way:

$$\max_{1 \leq i, j \leq N} |(\mathbf{\Lambda}^t)_{ij}| \leq \max_{0 \leq j \leq J-1} \binom{t}{t-j} \rho(\mathbf{R})^{t-j}$$

with  $\rho(\mathbf{R})$  the spectral radius of  $\mathbf{R}$  and  $J$  the maximum size of the associated Jordan blocks. Hence,  $\forall t > 0$

$$\max_{1 \leq i, j \leq N} |(\mathbf{\Lambda}^t)_{ij}| \leq t^{J-1} \rho(\mathbf{R})^{t-J+1} \quad (14)$$

When  $\mathbf{R}$  is diagonalizable,  $J = 1$ , and we get that

$$\max_{1 \leq i, j \leq N} |(\mathbf{\Lambda}^t)_{ij}| \leq \rho(\mathbf{R})^t \quad (\text{when } \mathbf{R} \text{ is diagonalizable}) \quad (15)$$

Putting together Eqs. (11), (13), (14), (15), and remarking that the subspace spanned by  $\mathbf{1}_{N^2} = \mathbf{1} \otimes \mathbf{1}$  is in the kernel of  $\mathbf{R}$ , we get that the size of the greatest Jordan block is  $\leq N - 1$ , hence the following lemma:

**Lemma 3.** *We have*

$$\mathbb{E}[\Psi_2(t)] = \mathcal{O}(t^{N-2} \rho(\mathbf{R})^t)$$

where  $\mathbf{R}$  is defined in Eq. (12) and where  $\rho(\mathbf{R})$  is the spectral radius of the matrix  $\mathbf{R}$ .

The next step of our analysis is to prove that the spectral radius  $\rho(\mathbf{R})$  is strictly less than 1 when Assumptions **(A1)**, **(A2)**, and **(B)** hold. Applying Theorem 5.6.12 of [15] on Eq. (11) proves that  $\rho(\mathbf{R}) < 1$  if and only if  $\mathbb{E}[\Xi(t)]$  converges to zero as  $t$  goes to infinity. Therefore our next objective is to prove that  $\mathbb{E}[\Xi(t)]$  converges to zero by using another way than the study of the spectral radius of  $\mathbf{R}$ .

Actually, one can find another linear recursion on  $\Xi(t)$  (different from the one exhibited in Eq. (10)). We get

$$\Xi(t) = \Xi(t-1) \cdot (\mathbf{K}(t) \otimes \mathbf{K}(t))$$

and, by taking the mathematical expectation given the past, we obtain

$$\mathbb{E}[\Xi(t)|\mathcal{F}_{t-1}] = \Xi(t-1) \cdot \mathbb{E}[\mathbf{K} \otimes \mathbf{K}].$$

Remarking that  $\Xi(t)\mathbf{1}_{N^2} = 0$ , we have for any vector  $\mathbf{v}$ ,

$$\mathbb{E}[\Xi(t)|\mathcal{F}_{t-1}] = \Xi(t-1) \cdot (\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - \mathbf{1}_{N^2}\mathbf{v}^T)$$

and then, for any vector  $\mathbf{v}$ ,

$$\mathbb{E}[\Xi(t)] = \Xi(0)\mathbf{S}_{\mathbf{v}}^t \tag{16}$$

with  $\mathbf{S}_{\mathbf{v}} = \mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - \mathbf{1}_{N^2}\mathbf{v}^T$  and  $\Xi(0) = (\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})$ .

By considering Eq. (16), it is straightforward to see that  $\mathbb{E}[\Xi(t)]$  converges to zero as  $t$  goes to infinity if there is a vector  $\mathbf{v}$  such that  $\rho(\mathbf{S}_{\mathbf{v}}) < 1$ . Notice that the recursion given in Eq. (16) is less “strong” than the one in Eq. (11) since it only leads to a sufficient condition instead of a necessary and sufficient condition. As  $\rho(\mathbf{S}_{\mathbf{v}}) < 1$  implies the convergence of  $\mathbb{E}[\Xi(t)]$  and as the convergence of  $\mathbb{E}[\Xi(t)]$  implies that  $\rho(\mathbf{R}) < 1$ , one thus can state the following Lemma:

**Lemma 4.** *If there is a vector  $\mathbf{v}$  such that  $\rho(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - \mathbf{1}_{N^2}\mathbf{v}^T) < 1$ , then  $\rho(\mathbf{R}) < 1$ .*

One of the most important result in the paper lies in the following Lemma in which we ensure that, under Assumptions **(A1)**, **(A2)**, and **(B)** there is a vector  $\mathbf{v}$  such that  $\rho(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - \mathbf{1}_{N^2}\mathbf{v}^T) < 1$  and thus  $\rho(\mathbf{R}) < 1$ .

**Lemma 5.** *If Assumptions **(A1)**, **(A2)**, **(B)** hold, there is a vector  $\mathbf{v}$  such that  $\rho(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - \mathbf{1}_{N^2}\mathbf{v}^T) < 1$ .*

*Proof:* Assumptions **(A1)**, **(A2)**, and **(B)** imply that

- i)  $\mathbb{E}[\mathbf{K} \otimes \mathbf{K}]$  is a non-negative matrix with a constant row sum equal to one (because of the row-stochasticity). According to Lemma 8.1.21 in [15], we have  $\rho(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}]) = 1$ .
- ii)  $\mathbb{E}[\mathbf{K} \otimes \mathbf{K}]$  is a primitive matrix (see **(B3)** in Lemma 1) which implies that there only is one eigenvalue of maximum modulus. This eigenvalue is thus equal to 1 and associated with the eigenvector  $\mathbf{1}_{N^2}$ .

By using the Jordan normal form and the simple multiplicity of the maximum eigenvalue (equal to 1), we know that i) there exists a vector  $\mathbf{v}_1$  equal to the left eigenvector corresponding to the eigenvalue 1, and ii) that the set of the eigenvalues of  $\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - \mathbf{1}_{N^2} \mathbf{v}_1^T = \mathbf{S}_{\mathbf{v}_1}$  are exactly the set of the eigenvalues of  $\mathbb{E}[\mathbf{K} \otimes \mathbf{K}]$  without the maximum one equal to 1. Indeed the maximum eigenvalue of  $\mathbb{E}[\mathbf{K} \otimes \mathbf{K}]$  has been removed by the vector  $\mathbf{1}_{N^2} \mathbf{v}_1^T$  and the associated eigenvector now belongs to the kernel of  $\mathbf{S}_{\mathbf{v}_1}$ . As a consequence, the modulus of the eigenvalues of  $\mathbf{S}_{\mathbf{v}_1}$  is strictly less than 1, *i.e.*,  $\rho(\mathbf{S}_{\mathbf{v}_1}) < 1$ . ■

Aggregating successively the results provided in Lemmas 5, 4, and 3 leads to the main result of this Section devoted to the analysis of  $\Psi_2(t)$ . Indeed, Lemma 5 ensures that there is a vector  $\mathbf{v}$  such that  $\rho(\mathbf{S}_{\mathbf{v}}) < 1$ , then Lemma 4 states that  $\rho(\mathbf{R}) < 1$ . Then, Lemma 3 concludes the proof for the next result.

**Proposition 2.** *Under Assumptions **(A1)**, **(A2)** and **(B)** holds, then*

$$\mathbb{E}[\Psi_2(t)] = \mathcal{O}(t^{N-2}e^{-\kappa t})$$

with  $\kappa = -\log(\rho(\mathbf{R})) > 0$ .

#### D. Final results

Thanks to the various intermediate Lemmas and Propositions provided above, we are now able to state the main Theorems of the paper. The first one deals with the determination of the necessary and sufficient conditions for Sum-Weight-like algorithms to converge. The second one gives us an insight on the decrease speed of the Squared Error (defined in Eq. (5)). In the meanwhile, we need the following lemma:

**Lemma 6.**  $\|\mathbf{x}(t) - x_{ave}\mathbf{1}\|_{\infty} = \max_i |x_i(t) - x_{ave}|$  is a non-increasing sequence with respect to  $t$ .

*Proof:* One can remark that, at time  $t + 1$ , we have

$$\begin{aligned} \forall j, x_j(t+1) &= \frac{\sum_{i=1}^N (\mathbf{K})_{ij} s_i(t)}{\sum_{i=1}^N (\mathbf{K})_{ij} w_i(t)} \\ &= \sum_{i=1}^N \left( \frac{(\mathbf{K})_{ij} w_i(t)}{\sum_{\ell=1}^N (\mathbf{K})_{\ell j} w_{\ell}(t)} \right) x_i(t) \end{aligned}$$

where  $\mathbf{K}$  corresponds to any matrix in  $\mathcal{K}$ . So  $x_j(t+1)$  is a center of mass of  $(x_i(t))_{i=1,\dots,N}$ . Therefore,  $\forall j \in \{1, \dots, N\}$ ,

$$\begin{aligned} |x_j(t+1) - x_{ave}| &\leq \sum_{i=1}^N \left( \frac{(\mathbf{K})_{ij} w_i(t)}{\sum_{\ell=1}^N (\mathbf{K})_{\ell j} w_\ell(t)} \right) |x_i(t) - x_{ave}| \\ &\leq \max_i |x_i(t) - x_{ave}|. \end{aligned}$$

■

*1) Result on the convergence:* Let us consider that Assumption **(B)** does not hold. Thanks to **(B1)** in Lemma 1, this is equivalent to  $\exists(k, l) \in N^2$  such that  $\forall T, \mathbf{P}(T)_{k,l} = 0$ . Let us take  $\mathbf{x}(0)$  equal to the canonical vector composed by a 1 at the  $k$ -th position and 0 elsewhere. Then for any  $t > 0$ ,  $x_l(t) = 0$  which is different from  $x_{ave} = 1/N$ . Consequently, the algorithm does not converge to the true consensus for any initial measurement. So if the Sum-Weight algorithm converges almost surely to the true consensus for any initial vector  $\mathbf{x}(0)$  then Assumption **(B)** holds.

Let us now assume that Assumption **(B)** holds. Using Markov's inequality along with Result 2, we have a finite  $K$  such that for any  $\delta > 0$ ,

$$\begin{aligned} \sum_{t>0} \mathbb{P}[|\Psi_2(t)| > \delta] &\leq \frac{1}{\delta} \sum_{t>0} \mathbb{E}[|\Psi_2(t)|] \\ &\leq \frac{1}{\delta} K \sum_{t>0} t^{N-2} e^{-\kappa t} < \infty. \end{aligned}$$

Consequently, Borel-Cantelli's Lemma leads to the almost sure convergence of  $\Psi_2(t)$  to zero. In addition, the random variables  $(\tau_n)_{n>0}$  provided in the statement of Proposition 1 converge to infinity with probability one, hence  $\Psi_2(\tau_n) \rightarrow 0$  almost surely. Since  $\Psi_1(\tau_n)$  is bounded,  $\Psi_1(\tau_n)\Psi_2(\tau_n) \xrightarrow{n \rightarrow \infty} 0$  almost surely. According to Lemma 6,  $\|\mathbf{x}(t) - x_{ave}\mathbf{1}\|_\infty$  is a nonincreasing nonnegative sequence verifying  $\|\mathbf{x}(t) - x_{ave}\mathbf{1}\|_\infty \leq \Psi_1(t)\Psi_2(t)$ , as there is converging subsequence with limit 0, the sequence itself converges to the same limit which implies the following theorem.

**Theorem 1.** *Under Assumptions **(A1)** and **(A2)**,  $\mathbf{x}(t)$  converges almost surely to the average consensus  $x_{ave}\mathbf{1}$  for any  $\mathbf{x}(0)$ , if and only if Assumption **(B)** holds.*

We have additional result on another type of convergence for  $\mathbf{x}(t)$ . As  $\|\mathbf{x}(t) - x_{ave}\mathbf{1}\|_\infty$  is a non-increasing sequence, we have, for any  $t$ ,  $\|\mathbf{x}(t) - x_{ave}\mathbf{1}\|_\infty \leq \|\mathbf{x}(0) - x_{ave}\mathbf{1}\|_\infty$  which implies that  $\mathbf{x}(t)$  is bounded for any  $t > 0$ . As a consequence, according to [16], since  $\mathbf{x}(t)$  also converges almost surely to  $x_{ave}\mathbf{1}$ , we know that  $\mathbf{x}(t)$  converges to  $x_{ave}\mathbf{1}$  in  $L^p$  for any positive integer  $p$ . The convergence of the mean squared error of  $\mathbf{x}(t)$  thus corresponds to the case  $p = 2$ .

**Corollary 1.** *If  $\mathbf{x}(t)$  converges almost surely to the average consensus  $x_{ave}\mathbf{1}$  then the mean squared error (MSE) converges to zero.*

2) *Result on the convergence speed:* The next result on the convergence speed corresponds to the main challenge and novelty of the paper. Except in [12] for a very specific case (cf. Section V-A for more details), our paper provides the first general results about the theoretical convergence speed for the squared error of the Sum-Weight like algorithms. For the sake of this theorem we introduce the following notation: given two sequences of random variables  $(X_n)_{n>0}$  and  $(Y_n)_{n>0}$ , we will say that  $X_n = o_{a.s.}(Y_n)$  if  $X_n/Y_n \rightarrow 0$  almost surely.

**Theorem 2.** *Under Assumptions (A1), (A2), and (B), the squared error (SE) is non-increasing. Furthermore, it is bounded by an exponentially decreasing function as follows*

$$\text{SE}(\tau_n) = \|\mathbf{x}(\tau_n) - x_{ave}\mathbf{1}\|_2^2 = o_{a.s.}(\tau_n^N e^{-\kappa\tau_n})$$

with  $\kappa = -\log(\rho((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})) \mathbb{E}[\mathbf{K} \otimes \mathbf{K}]) > 0$  and  $\tau_n = \sum_{i=1}^n \Delta_i$  as defined in Proposition 1.

This result tells us that the slope of  $\log(\text{SE}(t))$  is lower-bounded by  $\kappa$  infinitely often which provides us a good insight about the asymptotic behavior of  $\mathbf{x}(t)$ . Indeed, the squared error will vanish exponentially and we have derived a lower bound for this speed. We believe this result is new as it may foretell any algorithm speed. The particular behavior of the weights variables in this very general setting does not enable us to provide a clearer result about the mean squared error; however for some particular algorithms (e.g. single-variate ones) this derivation is possible (see Section V for more details). The authors would like to draw the reader's attention to the fact that the main contribution of the paper lies in the exponential decrease constant  $\kappa$ .

*Proof:* To prove this result we will once more use the decomposition of the squared error introduced in Eq. (6). We know from Proposition 2 that  $\mathbb{E}[t^{-N} e^{\kappa t} \Psi_2(t)] = \mathcal{O}(t^{-2})$ . By Markov's inequality and Borel-Cantelli's lemma,

$$t^{-N} e^{\kappa t} \Psi_2(t) \xrightarrow[t \rightarrow \infty]{} 0 \quad \text{almost surely.}$$

Composing with the  $(\tau_n)_{n>0}$ , we get

$$\tau_n^{-N} e^{\kappa \tau_n} \Psi_2(\tau_n) \xrightarrow[n \rightarrow \infty]{} 0 \quad \text{almost surely.}$$

Since  $\exists C, \forall n > 0, \Psi_1(\tau_n) \leq C$ , we get the claimed result. ■

#### IV. PROPOSED ALGORITHMS

In Subsection IV-A, we propose a new Sum-Weight-like algorithm using the broadcast nature of the wireless channel which converges and offers remarkable performance. This algorithm is hereafter called *Broadcast-Weighted Gossip (BWGossip)*. In Subsection IV-B, a new distributed management of the nodes' clocks which can improve averaging algorithms is proposed. Finally, Subsection IV-C provides an extension of this work to the distributed sum computation.

##### A. BWGossip algorithm

Remarking i) that the broadcast nature of the wireless channel was often not taken into account in the distributed estimation algorithms (apart in [9] but this algorithm does not converge to the average) and ii) that information propagation is much faster while broadcasting compared to pairwise exchanges [17], we propose an algorithm taking into account the broadcast nature of the wireless channel. At each global clock tick, it simply consists in uniformly choosing a sensor that broadcasts its pair of values in an appropriate way; then, the receiving sensors add their received pair of values to their current one. A more algorithmic formulation is presented below.

---

##### Algorithm 1 BWGossip

---

When the sensor  $i$  wakes up (at global time  $t$ ):

- The sensor  $i$  broadcasts  $\left( \frac{s_i(t)}{|\mathcal{N}_i|+1}; \frac{w_i(t)}{|\mathcal{N}_i|+1} \right)$
  - The sensors of the neighborhood  $\mathcal{N}_i$  update :  $\forall j \in \mathcal{N}_i, \begin{cases} s_j(t+1) = s_j(t) + \frac{s_i(t)}{|\mathcal{N}_i|+1} \\ w_j(t+1) = w_j(t) + \frac{w_i(t)}{|\mathcal{N}_i|+1} \end{cases}$
  - The sensor  $i$  updates :  $\begin{cases} s_i(t+1) = \frac{s_i(t)}{|\mathcal{N}_i|+1} \\ w_i(t+1) = \frac{w_i(t)}{|\mathcal{N}_i|+1} \end{cases}$
- 

According to this formulation, the update matrix  $\mathbf{K}_i$  associated with the action of the  $i$ -th sensor takes the following form

$$\begin{aligned} \mathbf{K}_i &= \mathbf{I} - e_i e_i^T + e_i e_i^T [(\mathbf{I} + \mathbf{D})^{-1} (\mathbf{A} + \mathbf{I})] \\ &= \mathbf{I} - e_i e_i^T (\mathbf{I} + \mathbf{D})^{-1} \mathbf{L} \end{aligned} \tag{17}$$

with  $e_i$  the  $i$ -th canonical vector. Clearly, the update matrices satisfy the Assumptions (A1) and (A2).

Thanks to Eq. (17) and recalling that  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , we obtain that

$$\begin{aligned}\mathbb{E}[\mathbf{K}] &= \mathbf{I} - \frac{1}{N}(\mathbf{I} + \mathbf{D})^{-1}\mathbf{L} \\ &= \frac{N-1}{N}\mathbf{I} + (\mathbf{I} + \mathbf{D})^{-1}(\mathbf{I} + \mathbf{A}).\end{aligned}$$

As all the involved matrices are non-negative, we have  $(\mathbf{I} + \mathbf{D})^{-1}(\mathbf{I} + \mathbf{A}) \geq (\mathbf{I} + \mathbf{A}) / ((d_{max} + 1)N)$ .

As a consequence, we have

$$\mathbb{E}[\mathbf{K}] \geq \frac{1}{(d_{max} + 1)N}(\mathbf{I} + \mathbf{A}) \geq 0.$$

Since  $\mathbf{A}$  is the adjacency matrix of a connected graph,  $\exists m > 0, (\mathbf{I} + \mathbf{A})^m > 0$ . Hence, for the same  $m$ ,  $\mathbb{E}[\mathbf{K}]^m \geq 1/(d_{max}N + N)^m(\mathbf{I} + \mathbf{A})^m > 0$ , which implies that  $\mathbb{E}[\mathbf{K}]$  is a primitive matrix. Applying Lemma 1 enables us to prove that Assumption **(B)** also holds.

Hence, Theorem 1 states that the BWGossip algorithm converges almost surely to the average consensus and Theorem 2 gives us an insight about the decrease speed of the squared error.

### B. Adaptation to smart clock management

So far, all the Poisson coefficients of the clocks were identical. This means that all sensors were waking up uniformly and independently from their past actions. Intuitively, it would be more logical that a sensor talking a lot became less active during a long period.

Another advantage of the Sum-Weight algorithms is the knowledge of how much a sensor talks compared to the others which is a useful information. Actually, each sensor knows whether it talks frequently or not (without additional cost) through its own weight value because when a sensor talks, its weight decreases and conversely when it receives information, its weight increases. Therefore, our idea is to control the Poisson coefficient of each sensor with respect to its weight.

We thus propose to consider the following rule for each Poisson coefficient

$$\forall i \in V, \quad \lambda_i(t) = \alpha + (1 - \alpha)w_i(t) \quad (18)$$

where  $\alpha \in (0, 1)$  is a tuning coefficient.

Notice that the global clock remains unchanged since  $\forall t > 0, \sum_{i=1}^N \lambda_i(t) = N$ . Keeping the global message exchange rate unchanged, the clock rates of each sensor are improved. The complexity of the algorithm is the same because the sensor whose weight changes has just to launch a Poisson clock.

Even if the convergence and the convergence speed with clock improvement have not been formally established, our simulations with the BWGossip algorithm (see Fig. 2) show that it seems to also converge exponentially to the average more quickly if  $\alpha$  is well chosen.



### C. Distributed estimation of the sum

In some cases, distributively computing the sum of the initial values is very interesting. For example, in the case of signal detection, the Log Likelihood Ratio (LLR) of a set of sensors is separable into the sum of the LLRs of the sensors. Hence, in order to perform a signal detection test based on the information of the whole network (using a Generalized LLR Test for instance), every sensor needs to estimate the sum of the LLRs computed by the sensors.

An estimate of the sum can be trivially obtained by multiplying the average estimate by the number of sensors which might not be available at any sensor. Another interest of the Sum-Weight scheme is that the initialization of the weights of the sensors enables us to compute different functions related to the average. Intuitively, as the sum of the  $\mathbf{s}(t)$  and  $\mathbf{w}(t)$  vectors are conserved through time and the convergence to a consensus is guaranteed by the assumptions on the update matrices, we get that the sensors will converge to  $\sum_i s_i(0)/\sum_i w_i(0)$ . This is obviously equal to the average  $1/N \sum_i x_i(0)$  with the initialisation of Eq. (1).

Now, if a sensor wants to trigger a estimation of the sum through the network, it simply sets its weight to 1 and sends a starting signal to the other nodes which set their weights to 0. Mathematically, we then have the following initialization after sensor  $i$  triggers the algorithm

$$\begin{cases} \mathbf{s}(0) = \mathbf{x}(0) \\ \mathbf{w}(0) = \mathbf{e}_i \end{cases}$$

where  $\mathbf{e}_i$  is the  $i$ -th canonical vector. In this setting, all Sum-Weight like algorithms converge exponentially to the sum of the initial value as all the theorems of the paper hold with only minor modifications in the proofs.

## V. COMPARISON WITH EXISTING WORKS

In this section, we will show that our results extend the works done previously in the literature. In Subsection V-A and V-B, we compare our results with existing papers dealing with the design and the analysis of the Sum-Weight like algorithms. In Subsection V-C, we will observe that our results can even be applied to the traditional framework of single-variate gossip algorithms.

### A. Comparison with Kempe's work

In the Kempe's work [12], the setup is quite different since the sensors' updates are synchronous, that is, at each time  $t$ , all the sensors send and update their values. Another important difference lies in the fact that the communication graph is assumed to be complete and to offer self-loops, i.e., each sensor

can communicate with any other one, including itself. The algorithm introduced in [12] is described in Algorithm 2.

---

**Algorithm 2** Push-Sum Algorithm [12]

---

At each time  $t$ , every sensor  $i$  activates:

- The sensor  $i$  chooses uniformly a node  $j_i(t)$  belonging to its neighborhood (including itself)
- The sensor  $i$  sends the pair  $(s_i(t)/2; w_i(t)/2)$  to  $j_i(t)$
- Let  $\mathcal{R}$  be the set of sensors that sent information to  $i$ . The sensor  $i$  updates:

$$\begin{cases} s_i(t+1) = s_i(t)/2 + \sum_{r \in \mathcal{R}} s_r(t)/2 \\ w_i(t+1) = w_i(t)/2 + \sum_{r \in \mathcal{R}} w_r(t)/2 \end{cases}$$


---

Consequently, at time  $t$ , the update matrix takes the following form

$$\mathbf{K}(t) = \frac{1}{2}\mathbf{I} + \frac{1}{2} \sum_{i=1}^N e_i e_{j_i(t)}^T \quad (19)$$

where the index  $j_i(t)$  is defined in Algorithm 2. Notice that the first term of the right hand side corresponds to the information kept by the sensor, while the second term corresponds to the information sent to the chosen sensor. Moreover, as each sensor selects uniformly its neighbor<sup>2</sup> (including itself), we obtain that

$$\mathbb{E}[\mathbf{K}] = \frac{1}{2}\mathbf{I} + \frac{1}{2}\mathbf{J}.$$

It is then easy to check that

- the (instantaneous) update matrices are non-negative and row-stochastic. In addition, they are chosen uniformly in a set of size  $N^N$ .
- the (instantaneous) update matrices have a strictly positive diagonal.
- $\mathbb{E}[\mathbf{K}] > 0$ , thus  $\mathbb{E}[\mathbf{K}]$  is a primitive matrix.

This proves that the Kempe's algorithm satisfies the assumptions **(A1)**, **(A2)** and **(B)**, and so it converges almost surely to the average consensus (which was also proven in [12]).

Let us now focus on the convergence speed of the Kempe's algorithm. We remind that the convergence speed is driven by  $\Psi_2(t)$  (denoted by  $\Phi_t$  in [12]). As this algorithm is synchronous and only applies on a complete communication graph, it is simple to obtain a recursion between  $\mathbb{E}[\Psi_2(t)|\Psi_2(t-1)]$  and  $\Psi_2(t-1)$ . Indeed, the approach given in the footnote of Section III-C can be applied. More precisely,

<sup>2</sup>as the graph is complete, this means, choosing one node uniformly in the graph.

the corresponding matrix  $\mathbf{M} = (\mathbf{I} - \mathbf{J}) \mathbb{E}[\mathbf{K}\mathbf{K}^T] (\mathbf{I} - \mathbf{J})$  is given in closed-form as (see Appendix B-A for details)

$$\mathbf{M} = (\mathbf{I} - \mathbf{J}) \mathbb{E}[\mathbf{K}\mathbf{K}^T] (\mathbf{I} - \mathbf{J}) = \left( \frac{1}{2} - \frac{1}{4N} \right) (\mathbf{I} - \mathbf{J}), \quad (20)$$

and then one can easily check<sup>3</sup> that

$$\mathbb{E}[\Psi_2(t)|\Psi_2(t-1)] = \left( \frac{1}{2} - \frac{1}{4N} \right) \Psi_2(t-1). \quad (21)$$

Moreover, thanks to Eq. (20), we have that  $\rho(\mathbf{M}) = (1/2 - 1/(4N)) < 1$  and thus the inequality in the above-mentioned footnote has been replaced with an equality and the spectral radius of  $\mathbf{M}$  is less than 1. Therefore, the true convergence speed is provided by  $\rho(\mathbf{M})$ . Comparing this previous convergence speed (obtained very easily in [12]) with the convergence speed bounds obtained in our paper is of great interest and will be done below. First of all we remind (see the footnote in Section III-C) that in the general case treated in our paper, it is impossible to find a recursion similar to Eq. (21) which justifies our alternative approach. Secondly, following the general alternative approach developed in this paper, we know that the matrix of interest is  $\mathbf{R} = ((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})) \cdot \mathbb{E}[\mathbf{K} \otimes \mathbf{K}]$  (see Proposition 2). After some computations (a detailed proof is available in Appendix B-B), we have that

$$\mathbf{R} = \frac{1}{4} (\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}) + \frac{N-1}{4N} \mathbf{v}\mathbf{v}^T \quad (22)$$

with  $\mathbf{v} = (1/\sqrt{N-1}) (\mathbf{u} - (1/N)\mathbf{1}_{N^2})$  and  $\mathbf{u} = \sum_{i=1}^N e_i \otimes e_i$ .

Consequently,  $\mathbf{R}$  is a linear combination of two following orthogonal projections:

- the first projection, generated by  $(\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})$ , is of rank  $N^2 - 2N + 1$ ,
- the second projection, generated by  $\mathbf{v}\mathbf{v}^T$ , is of rank 1.

As  $(\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})$  and  $\mathbf{v}\mathbf{v}^T$  are orthogonal projections, the vector space  $\mathbb{R}^{N^2}$  (on which the matrix  $\mathbf{R}$  is operating) can be decomposed into a direct sum of four subspaces:

- $\mathcal{S}_0 = \mathcal{Im}(\mathbf{v}\mathbf{v}^T) \cap \mathcal{Ker}((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}))$
- $\mathcal{S}_1 = \mathcal{Im}(\mathbf{v}\mathbf{v}^T) \cap \mathcal{Im}((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}))$
- $\mathcal{S}_2 = \mathcal{Ker}(\mathbf{v}\mathbf{v}^T) \cap \mathcal{Im}((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}))$
- $\mathcal{S}_3 = \mathcal{Ker}(\mathbf{v}\mathbf{v}^T) \cap \mathcal{Ker}((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}))$

As  $((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})) \mathbf{v} = \mathbf{v}$  (see Appendix B-B), we have  $\mathcal{S}_0 = \{0\}$ .

<sup>3</sup>Note that there is a typo in Lemma 2.3 of [12]. Indeed, the coefficient is  $(1/2 - 1/(2N))$  in [12] instead of  $(1/2 - 1/(4N))$ .

Moreover, according to Eq. (22), we obtain that

$$\mathbf{R}\mathbf{x} = \begin{cases} (\frac{1}{2} - \frac{1}{4N})\mathbf{x} & \forall \mathbf{x} \in \mathcal{S}_1 \\ \frac{1}{4}\mathbf{x} & \forall \mathbf{x} \in \mathcal{S}_2 \\ 0 & \forall \mathbf{x} \in \mathcal{S}_3 \end{cases}$$

As a consequence, the non-null eigenvalues of  $\mathbf{R}$  are  $1/4$  and  $(1/2 - 1/(4N))$  which implies that  $\rho(\mathbf{R}) = 1/2 - 1/(4N)$ . Hence, the convergence speed bound obtained by our general alternative approach developed in this paper provides the true convergence speed for the Kempe's algorithm [12].

### B. Comparison with Bénézit's algorithm

In [7], it has been shown that doing a multi-hop communication between sensors provides significant performance gain. However, the proposed algorithm relied on a single-variate algorithm. In order to ensure the convergence of this algorithm, the double-stochasticity of the matrix update is necessary which implies a feedback along the route. The feedback can suffer from link failure (due to high mobility in wireless networks). To counter-act this issue, Bénézit proposes to get rid of the feedback by using the Sum-Weight approach [13]. In this paper, the authors established a general convergence theorem close to ours. In contrast, they did not provide any result about convergence speed. It is worth noting that our convergence speed results can apply to the Bénézit's algorithm.

### C. Comparison with the single-variate algorithms

If the following additional assumption holds,

**(A3)** The matrices of  $\mathcal{K}$  are column-stochastic,

one can easily show that all the weights  $\mathbf{w}(t)$  remain constant and equal to  $\mathbf{1}$ , i.e.,

$$\begin{aligned} \forall t > 0, \quad \mathbf{w}(t)^T &= \mathbf{w}(0)^T \mathbf{P}(t) = \mathbf{1}^T \mathbf{P}(t) = \mathbf{1}^T \\ \text{and} \quad \mathbf{x}(t) &= \mathbf{s}(t) = \mathbf{K}(t)^T \mathbf{x}(t-1). \end{aligned}$$

Therefore, the single-variate algorithms ([18]) with double-stochastic update matrices such as the *Random Gossip* [3], [4], the *Geographic Gossip* [6] can surprisingly be cast into the Sum-Weight framework. Moreover as  $\Psi_1(t) = \|\mathbf{x}(0)\|_2^2$  because all the weights stay equal to 1, the proposed results about  $\Psi_2(t)$  (that is Section III-C) can be applied directly to the squared error for these algorithms.

Let us re-interpret the work of Boyd *et al.* [4] (especially their section 2) in the light of our results. In [4], it is stated that under doubly-stochastic update matrices  $\mathbf{K}(t)$ , the mean squared error at time  $t$  is

dominated by  $\rho(\mathbb{E}[\mathbf{K}\mathbf{K}^T] - (1/N)\mathbf{1}\mathbf{1}^T)^t$  and converges to 0 when  $t$  goes to infinity if

$$\rho\left(\mathbb{E}[\mathbf{K}] - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) < 1. \quad (23)$$

Since  $\mathbf{K}(t)$  is doubly-stochastic, one can remark that  $(\mathbf{I} - \mathbf{J}) \mathbb{E}[\mathbf{K}\mathbf{K}^T] (\mathbf{I} - \mathbf{J}) = \mathbb{E}[\mathbf{K}\mathbf{K}^T] - (1/N)\mathbf{1}\mathbf{1}^T$ . By following the approach developed in the footnote of Section III-C, we obtained directly the domination proven in [4]. Moreover, the condition corresponding to Eq. (23) actually implies Assumption **(B)**. Indeed, due to Eq. (23) and the double-stochasticity of  $\mathbf{K}(t)$ , one can remark that the maximum eigenvalue of  $\mathbb{E}[\mathbf{K}]$  is unique and equal to 1. Consequently,  $\mathbb{E}[\mathbf{K}]$  is primitive, and thus Assumption **(B)** holds (see Lemma 1). Furthermore, in [4] (see section II-B), it is stated that the condition corresponding to Eq. (23) is only a sufficient condition and that the necessary and sufficient condition is the following one

$$\rho\left(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - \frac{1}{N}\mathbf{1}_{N^2}\mathbf{1}_{N^2}^T\right) < 1 \quad (24)$$

which is exactly the same expression as that in Lemmas 4 and 5<sup>4</sup>. Along with the reasoning detailed in Section III-D1, these two lemmas prove that under assumptions **(A1)** and **(A2)**, the condition corresponding to Eq. (24) is eventually necessary and sufficient when assumption **(A3)** is also satisfied.

Moreover, according to Eq. (19) (in [4]) and Eq. (16) (in our paper), we know that the mean squared error at time  $t$  is upper bounded by  $-\kappa't$  with  $\kappa' = -\log(\rho(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - (1/N)\mathbf{1}_{N^2}\mathbf{1}_{N^2}^T)) > 0$ . However, as stated in Proposition 2, the logarithm of the squared error scales with  $-\kappa t$ . Though these two spectral radii are less 1 and so ensure the convergence,  $\rho((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}) \mathbb{E}[\mathbf{K} \otimes \mathbf{K}])$  (*i.e.*  $e^{-\kappa}$ ) exhibited in our paper is in general smaller than  $\rho(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] - (1/N)\mathbf{1}_{N^2}\mathbf{1}_{N^2}^T)$  (*i.e.*  $e^{-\kappa'}$ ) introduced in [4]. Hence, thanks to our approach, a tighter convergence speed bound has been derived. Numerical illustrations related to this statement are displayed on Fig. 4.

## VI. NUMERICAL RESULTS

In order to investigate the performance of distributed averaging algorithms over Wireless Sensor Networks, the use of Random Geometric Graphs (RGG) is commonly advocated. These graphs consist in uniformly placing  $N$  points in the unit square (representing the vertices of the future graph) then connecting those which are closer than a predefined distance  $r$ . A choice of  $r$  of the form  $\sqrt{r_0 \log(N)/N}$  with  $r_0 \in [1, \dots, 10]$  ensures connectedness with high probability when  $N$  becomes large and avoids complete graphs (see [19] for more details).

<sup>4</sup>Indeed, as the vector  $\mathbf{v}$  used in our formulation can be replaced with the left eigenvector corresponding to the eigenvalue 1 (see the proof of Lemma 5 for more details) which is proportional to  $\mathbf{1}$  here due to the double-stochasticity of the update matrices

In Fig. 1, we plot the empirical mean squared error versus time for different gossip algorithms: i) the *Random Gossip* [3] which is the reference algorithm in the literature; ii) the *Broadcast Gossip* introduced in [9] which uses the broadcasting abilities of the wireless channel but does not converge to the average; iii) the algorithm introduced by Franceschelli in [11] which uses a bivariate scheme and seems to converge (no convergence proof is provided in the paper); and iv) the proposed *BWGossip* algorithm. A Random Geometric Graphs with  $N = 100$  sensors and  $r_0 = 4$  has been considered. We remark that the *BWGossip* algorithm outperforms the existing algorithms without adding routing or any other kind of complexity.

In Fig. 2, we plot the empirical mean squared error for the *BWGossip* algorithm versus time with different clock tuning coefficients (see IV-B and Eq. (18) for more details). Compared to the algorithm without clock management ( $\alpha = 1$ ), the convergence is much faster at the beginning with  $\alpha = 0$  but the asymptotic rate is lower; with  $\alpha = 0.5$ , the performance is better than the *BWGossip* for any time.

In Fig. 3, we display the empirical convergence slope<sup>5</sup> and the associated lower-bound  $\kappa$  derived in Theorem 2 for the *BWGossip* algorithm versus the number of sensors  $N$ . Different Random Geometric Graphs with  $r_0 = 4$  have been considered. We observe a very good agreement between the empirical slope and the proposed lower bound. Consequently, our bound is very tight.

In Fig. 4, we display the empirical convergence slope, the associated lower-bound  $\kappa$ , and the bound given in [4] for the *Random Gossip* algorithm versus the number of sensors  $N$ . The proposed bound  $\kappa$  fits much better than the one proposed in [4]. Actually, the proposed bound matches very well the empirical slope (see Section V-C for more details).

Thanks to Fig. 5, we inspect the influence of link failures in the underlying communication graph on the *BWGossip* algorithm. We consider a Random Geographic Graph with 10 sensors and  $r_0 = 1$  onto which i.i.d. link failure events appear with probability  $p_e$ . In Fig. 5a, we plot the empirical mean squared error of the *BWGossip* versus time for different values of the edge failure probability  $p_e$ . As expected, we observe that the higher  $p_e$  the slower the convergence but the MSE still exponentially decreases. Then, in Fig. 5b, we plot the empirical convergence slope and the associated bound  $\kappa$  for different link failure probabilities. Here,  $\kappa$  is computed according to a modified matrix set taking into account the link failures through different update matrices. We remark a very good fitting between our lower bound and the simulated results. Consequently, computing  $\kappa$  on the matrix set including the link failures enables us to predict very well the convergence speed in this context.

<sup>5</sup>this slope has been obtained by linear regression on the logarithm of the empirical mean squared error. This regression makes sense since, for inspected algorithms, the mean squared error in log scale is almost linear for  $t$  large enough as seen in Fig. 1.

## VII. CONCLUSION

In this paper, we have analyzed the convergence of the Sum-Weight-like algorithms (relying on two variables rather than one) for distributed averaging in a Wireless Sensor Network. We especially give a very precise insight on the convergence speed of the squared error for such algorithms. In addition, we proposed a particular Sum-Weight-like algorithm taking full advantage of the broadcast nature of the wireless channel. We observed that this algorithm significantly outperforms the existing ones.

## APPENDIX A

### PROOF OF LEMMA 1

**(B)  $\Rightarrow$  (B1)** Let denote by  $\mathbf{K}^{(u,v)}$  a matrix of  $\mathcal{K}$  whose  $(u, v)$ -th coefficient is positive. As the graph associated with  $\mathbb{E}[\mathbf{K}]$  is connected, then for all couples of nodes  $(i, j)$ , there is a path of finite length  $L_{ij} < N$  from  $i$  to  $j$ :  $(i = u_1, \dots, u_{L_{ij}} = j)$ . Consequently, the matrix  $\mathbf{K}^{i \rightarrow j} = \mathbf{K}^{(u_1, u_2)} \mathbf{K}^{(u_2, u_3)} \dots \mathbf{K}^{(u_{L_{ij}-1}, u_{L_{ij}})}$  verifies:  $(\mathbf{K}^{i \rightarrow j})_{i,j} > 0$  which gives us a realization of  $\mathbf{P}(L_{ij})$  verifying  $(\mathbf{P}(L_{ij}))_{i,j} > 0$ .

**(B1)  $\Rightarrow$  (B2)** Let us take  $L = \sum_{i,j=1}^N L_{ij} < 2N^2$ . Since each matrix has a positive diagonal according to Assumption **(A2)** then  $\prod_{i,j=1}^N \mathbf{K}^{i \rightarrow j}$  is a possible realization of  $\mathbf{P}(L)$  of strictly positive probability which is a positive matrix.

**(B2)  $\Rightarrow$  (B3)** If there is a  $L < 2N^2$  and a realization  $\mathbf{p}$  of  $\mathbf{P}(L)$  so that  $\mathbb{P}[\mathbf{P}(L) = \mathbf{p}] > 0$  and  $\mathbf{p} > 0$ , then  $\mathbf{p} \otimes \mathbf{p}$  is also positive. Since  $(\mathbf{A} \otimes \mathbf{B}).(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$  for any matrices  $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$  with the appropriate dimensions,

$$(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}])^L = \left( \sum_{i=1}^M p_i \mathbf{K}_i \otimes \mathbf{K}_i \right)^L \geq \mathbb{P}[\mathbf{P}(L) = \mathbf{p}]. \mathbf{p} \otimes \mathbf{p} > 0.$$

Hence,  $\mathbb{E}[\mathbf{K} \otimes \mathbf{K}]$  is a primitive matrix.

**(B3)  $\Rightarrow$  (B)** First, we will calculate  $\mathbb{E}[\mathbf{K}] \otimes \mathbb{E}[\mathbf{K}]$  with respect to  $\mathbb{E}[\mathbf{K} \otimes \mathbf{K}]$ . So,

$$\begin{aligned} \mathbb{E}[\mathbf{K}] \otimes \mathbb{E}[\mathbf{K}] &= \sum_{i=1}^M \sum_{j=1}^M p_i p_j \mathbf{K}_i \otimes \mathbf{K}_j \\ &\geq \sum_{i=1}^M p_i^2 \mathbf{K}_i \otimes \mathbf{K}_i \\ &\geq (\min_j p_j) \cdot \sum_{i=1}^M p_i \mathbf{K}_i \otimes \mathbf{K}_i = (\min_j p_j) \cdot \mathbb{E}[\mathbf{K} \otimes \mathbf{K}] \end{aligned}$$

Hence as it exists  $k$  such that  $(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}])^k > 0$ , then  $(\mathbb{E}[\mathbf{K}])^k > 0$  so the primitivity of  $\mathbb{E}[\mathbf{K}]$  is proven.

## APPENDIX B

## DERIVATIONS RELATED TO SECTION V

## A. Derivations for Eq. (20)

According to Eq. (19), we have easily that

$$\mathbf{K}(t)\mathbf{K}(t)^T = \frac{1}{4}\mathbf{I} + \frac{1}{4}\sum_{i=1}^N e_i e_{j_i(t)}^T + \frac{1}{4}\sum_{i=1}^N e_{j_i(t)} e_i^T + \frac{1}{4}\sum_{i=1}^N \sum_{i'=1}^N e_i e_{j_i(t)}^T e_{j_{i'}(t)} e_{i'}^T$$

By remarking that  $e_j^T e_j = 1$ , we have

$$\mathbf{K}(t)\mathbf{K}(t)^T = \frac{1}{2}\mathbf{I} + \frac{1}{4}\sum_{i=1}^N e_i e_{j_i(t)}^T + \frac{1}{4}\sum_{i=1}^N e_{j_i(t)} e_i^T + \frac{1}{4}\sum_{i=1}^N \sum_{\substack{i'=1 \\ i' \neq i}}^N e_i e_{j_i(t)}^T e_{j_{i'}(t)} e_{i'}^T$$

The randomness in  $\mathbf{K}(t)\mathbf{K}(t)^T$  is only due to the choice of the nodes  $j_i(t)$  for  $i = \{1, \dots, N\}$ . Therefore, each  $j_i(t)$  will be modeled by a random variable  $\ell(i)$  (independent of  $t$ ). The random variables  $\{\ell(i)\}_{i=1, \dots, N}$  are i.i.d. and are uniformly distributed over  $\{1, \dots, N\}$ . As a consequence, we obtain

$$\mathbb{E}[\mathbf{K}\mathbf{K}^T] = \frac{1}{2}\mathbf{I} + \frac{1}{4}\sum_{i=1}^N e_i \left( \frac{1}{N} \sum_{k=1}^N e_k^T \right) + \frac{1}{4}\sum_{i=1}^N \left( \frac{1}{N} \sum_{k=1}^N e_k \right) e_i^T + \frac{1}{4}\sum_{i=1}^N \sum_{\substack{i'=1 \\ i' \neq i}}^N e_i \left( \frac{1}{N^2} \sum_{k, k'=1}^N e_k^T e_{k'} \right) e_{i'}^T$$

By remarking that  $e_k^T e_{k'} = 0$  as soon as  $k \neq k'$ , we have  $\sum_{k, k'=1}^N e_k^T e_{k'} = N$ . Furthermore,

$$\text{as } \sum_{k=1}^N e_k = \mathbf{1} \quad \text{and} \quad \sum_{i=1}^N \sum_{\substack{i'=1 \\ i' \neq i}}^N e_i e_{i'}^T = \mathbf{1}\mathbf{1}^T - \mathbf{I}$$

$$\text{we obtain } \mathbb{E}[\mathbf{K}\mathbf{K}^T] = \left( \frac{1}{2} - \frac{1}{4N} \right) \mathbf{I} + \frac{3}{4} \mathbf{J}$$

It is then straightforward to obtain Eq. (20).

## B. Derivations for Eq. (22)

Once again, according to Eq. (19), we have easily that

$$\begin{aligned} \mathbf{K}(t) \otimes \mathbf{K}(t) &= \frac{1}{4}\mathbf{I} \otimes \mathbf{I} + \frac{1}{4} \left( \sum_{i=1}^N e_i e_{j_i(t)}^T \right) \otimes \mathbf{I} + \frac{1}{4}\mathbf{I} \otimes \left( \sum_{i=1}^N e_i e_{j_i(t)}^T \right) \\ &+ \underbrace{\frac{1}{4} \left( \sum_{i=1}^N e_i e_{j_i(t)}^T \right) \otimes \left( \sum_{i'=1}^N e_{i'} e_{j_{i'}(t)}^T \right)}_{\xi} \end{aligned} \quad (25)$$

Using the same technique as in Appendix B-A, we obtain that

$$\mathbb{E} \left[ \sum_{i=1}^N e_i e_{j_i(t)}^T \right] = \sum_{i=1}^N e_i \left( \frac{1}{N} \sum_{k=1}^N e_k \right) = \mathbf{J} \quad (26)$$



Thus, it just remains to evaluate  $\mathbb{E}[\xi]$ . Let us first remark that

$$\xi = \sum_{i=1}^N \sum_{\substack{i'=1 \\ i' \neq i}}^N e_i e_{j_i(t)}^T \otimes e_{i'} e_{j_{i'}(t)}^T + \sum_{i=1}^N e_i e_{j_i(t)}^T \otimes e_i e_{j_i(t)}^T$$

As a consequence, we have

$$\begin{aligned} \mathbb{E}[\xi] &= \frac{1}{N^2} \sum_{i=1}^N \sum_{\substack{i'=1 \\ i' \neq i}}^N \sum_{k=1}^N \sum_{k'=1}^N e_i e_k^T \otimes e_{i'} e_{k'}^T + \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^N e_i e_k^T \otimes e_i e_k^T \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N \sum_{k=1}^N \sum_{k'=1}^N e_i e_k^T \otimes e_{i'} e_{k'}^T + \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^N e_i e_k^T \otimes e_i e_k^T - \frac{1}{N^2} \sum_{i=1}^N \sum_{k=1}^N \sum_{k'=1}^N e_i e_k^T \otimes e_i e_{k'}^T \end{aligned}$$

Using the well-known result on Kronecker product ( $(\mathbf{AB}) \otimes (\mathbf{CD}) = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \otimes \mathbf{D})$  for four matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  with appropriate sizes), we have

$$\mathbb{E}[\xi] = \mathbf{J} \otimes \mathbf{J} + \frac{1}{N} \mathbf{u} \mathbf{u}^T - \frac{1}{N^2} \mathbf{u} \mathbf{1}_{N^2}^T. \quad (27)$$

Putting Eqs. (26)-(27) into Eq. (25), we get

$$\mathbb{E}[\mathbf{K} \otimes \mathbf{K}] = \frac{1}{4} \mathbf{I} \otimes \mathbf{I} + \frac{1}{4} \mathbf{J} \otimes \mathbf{I} + \frac{1}{4} \mathbf{I} \otimes \mathbf{J} + \frac{1}{4} \mathbf{J} \otimes \mathbf{J} + \frac{1}{4N} \mathbf{u} \mathbf{u}^T - \frac{1}{4N^2} \mathbf{u} \mathbf{1}_{N^2}^T.$$

Before going further, let us remark that

$$\begin{aligned} ((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})) \mathbf{u} &= \sum_{i=1}^N (e_i - \frac{1}{N} \mathbf{1} \mathbf{1}^T e_i) \otimes (e_i - \frac{1}{N} \mathbf{1} \mathbf{1}^T e_i) \\ &= \sum_{i=1}^N e_i \otimes e_i - \sum_{i=1}^N (e_i \otimes \frac{1}{N} \mathbf{1}) - \sum_{i=1}^N (\frac{1}{N} \mathbf{1} \otimes e_i) + \frac{1}{N^2} \sum_{i=1}^N \mathbf{1} \otimes \mathbf{1} \\ &= \mathbf{u} - \frac{1}{N} \mathbf{1}_{N^2}. \end{aligned} \quad (28)$$

As a consequence, we have

$$\begin{aligned} \mathbf{R} &= ((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})) \cdot \mathbb{E}[\mathbf{K} \otimes \mathbf{K}] \\ &= \frac{1}{4} (\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}) + \frac{1}{4N} \left( \mathbf{u} - \frac{1}{N} \mathbf{1}_{N^2} \right) \mathbf{u}^T - \frac{1}{4N^2} \left( \mathbf{u} - \frac{1}{N} \mathbf{1}_{N^2} \right) \mathbf{1}_{N^2}^T \\ &= \frac{1}{4} (\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J}) + \frac{1}{4N} \mathbf{u} \mathbf{u}^T - \frac{1}{4N^2} \mathbf{1}_{N^2} \mathbf{u}^T - \frac{1}{4N^2} \mathbf{u} \mathbf{1}_{N^2}^T + \frac{1}{4N} \mathbf{J} \otimes \mathbf{J} \end{aligned}$$

Let us remind  $\mathbf{v} = \frac{1}{\sqrt{N-1}} \left( \mathbf{u} - \frac{1}{N} \mathbf{1}_{N^2} \right)$ . Thanks to Eq. (28), we have

$$\mathbf{v} \mathbf{v}^T = \frac{1}{N-1} \left( \mathbf{u} \mathbf{u}^T - \frac{1}{N} \mathbf{1}_{N^2} \mathbf{u}^T - \frac{1}{N} \mathbf{u} \mathbf{1}_{N^2}^T + \mathbf{J} \otimes \mathbf{J} \right)$$

which straightforwardly leads to Eq. (22).

In addition, note that using Eq. (28), we have  $((\mathbf{I} - \mathbf{J}) \otimes (\mathbf{I} - \mathbf{J})) \mathbf{v} = \mathbf{v}$ .

## REFERENCES

- [1] F. Iutzeler, P. Ciblat, W. Hachem, and J. Jakubowicz, "A new broadcast based distributed averaging algorithm over Wireless Sensor Networks," in *Proc. 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 3117–3120.
- [2] J. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, M. I. T., Dept. of Electrical Engineering and Computer Science, 1984.
- [3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Analysis and optimization of randomized gossip algorithms," in *Proc. 43rd IEEE Conference on Decision and Control (CDC)*, vol. 5, 2004, pp. 5310–5315.
- [4] —, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [5] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip Algorithms for Distributed Signal Processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.
- [6] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic Gossip: Efficient Averaging for Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1205–1216, 2008.
- [7] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli, "Order-Optimal Consensus Through Randomized Path Averaging," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5150–5167, 2010.
- [8] D. Ustebay, B. N. Oreshkin, M. J. Coates, and M. G. Rabbat, "Greedy Gossip With Eavesdropping," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3765–3776, 2010.
- [9] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast Gossip Algorithms for Consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748–2761, 2009.
- [10] B. Nazer, A. G. Dimakis, and M. Gastpar, "Neighborhood gossip: Concurrent averaging through local interference," in *Proc. 34th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 3657–3660.
- [11] M. Franceschelli, A. Giua, and C. Seatzu, "Distributed Averaging in Sensor Networks Based on Broadcast Gossip Algorithms," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 808–817, 2011.
- [12] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. 44th Annual IEEE Symposium on the Foundations of Computer Science (FOCS)*, 2003, pp. 482–491.
- [13] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted Gossip: Distributed Averaging using non-doubly stochastic matrices," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2010, pp. 1753–1757.
- [14] N. Biggs, *Algebraic Graph Theory*, Cambridge Univ Press, 1993.
- [15] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, 2005.
- [16] A. W. Van der Vaart, *Asymptotic Statistics*, Cambridge University Press, 2000.
- [17] F. Iutzeler, P. Ciblat, and J. Jakubowicz, "Analysis of max-consensus algorithms in wireless channels," *IEEE Transactions on Signal Processing*, vol. PP, no. 99, p. 1, 2012.
- [18] A. Tahbaz-Salehi and A. Jadbabaie, "A Necessary and Sufficient Condition for Consensus Over Random Networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 791–795, 2008.
- [19] M. Penrose, *Random geometric graphs*, Oxford University Press, 2003.
- [20] F. Bullo, J. Cortés, and S. Martinez, "Introduction to distributed algorithms," in *Proc. 47th IEEE Conference on Decision and Control (CDC)*, 2008.
- [21] R. Varga, *Matrix iterative analysis*, Springer, 2010.

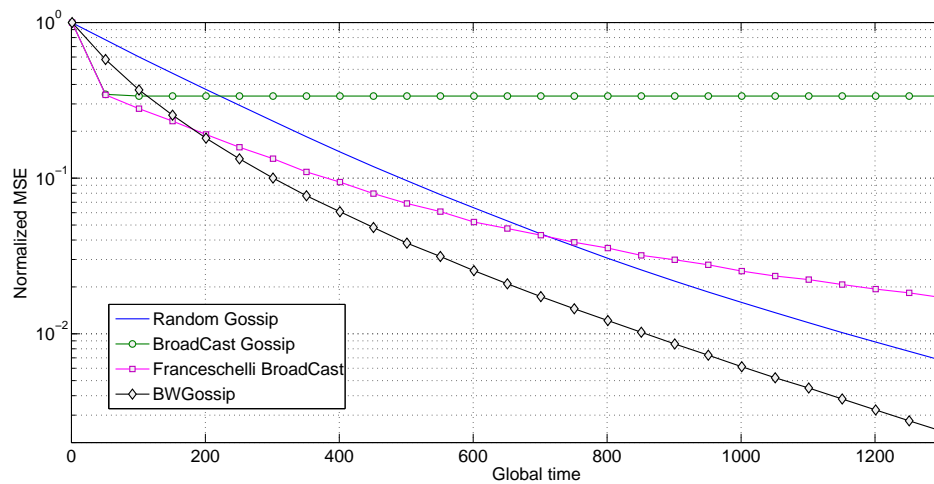


Fig. 1: Mean squared error of the *BWGossip* and other famous algorithms versus time.

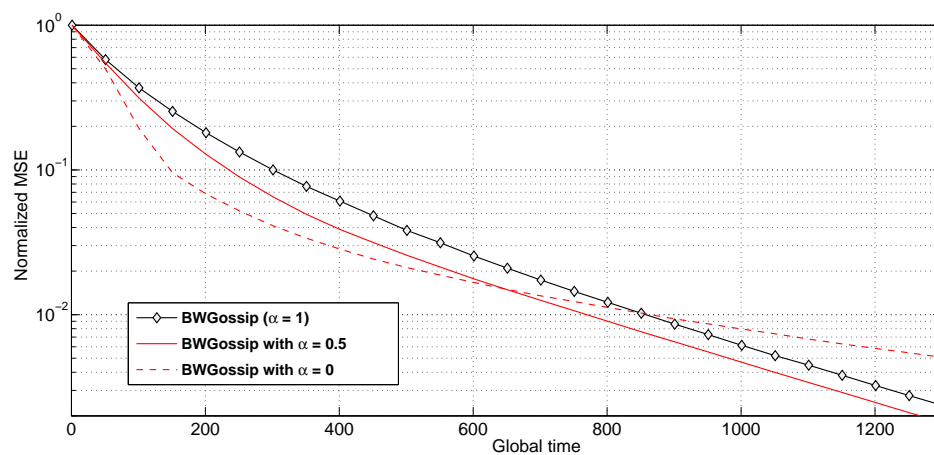


Fig. 2: Mean squared error of the *BWGossip* versus time for different clock management schemes.

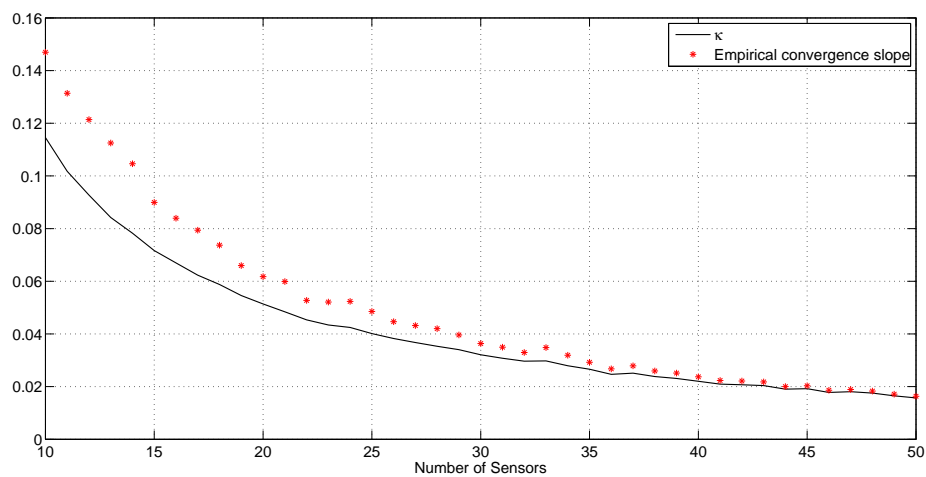


Fig. 3: Empirical convergence slope of the *BWGossip* and the associated lower bound  $\kappa$ .

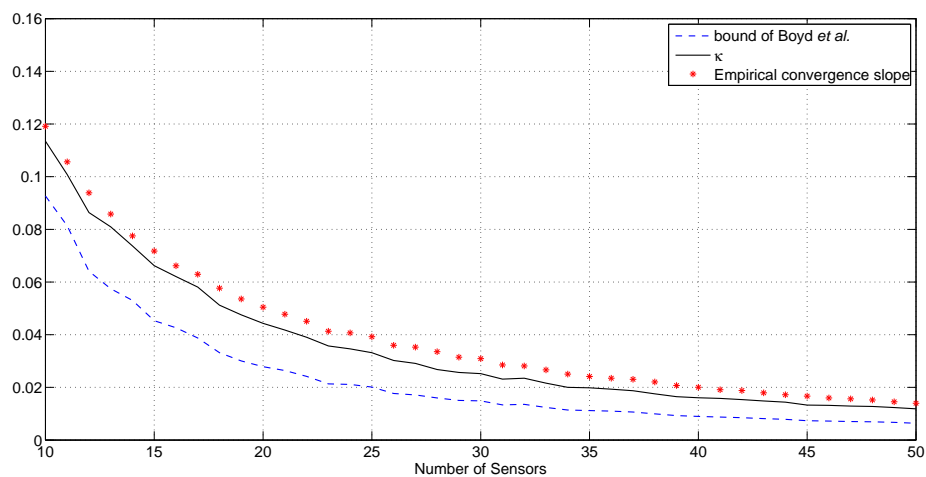
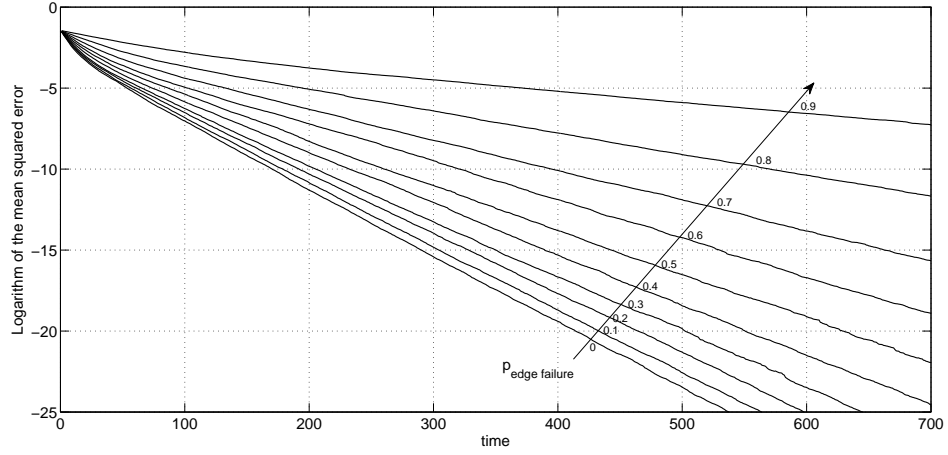
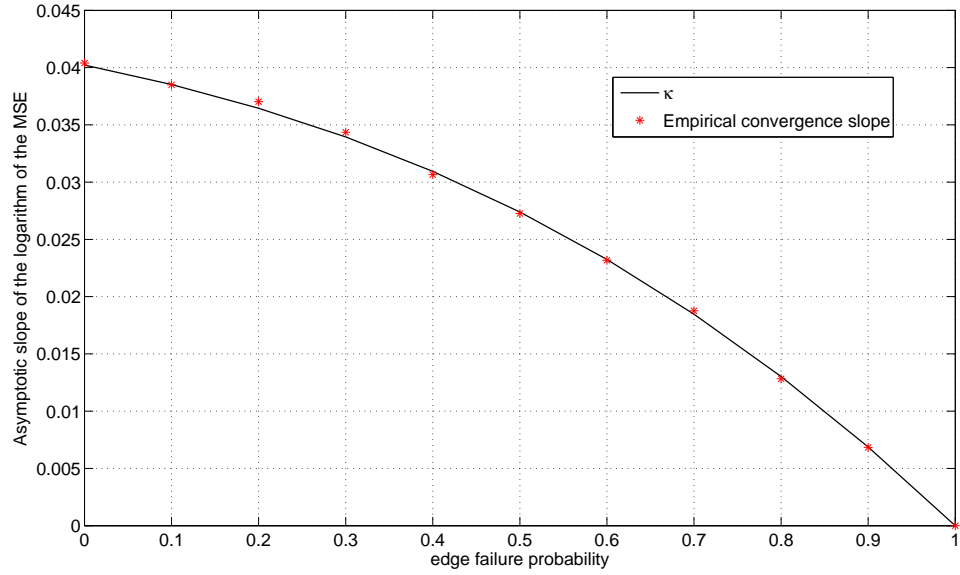


Fig. 4: Empirical convergence slope of the *Random Gossip*, the associated lower bound  $\kappa$ , and the bound given in [4].



(a) Mean squared error versus time for different link failure probabilities.

(b) Empirical convergence slope and the associated lower bound  $\kappa$  versus link failure probabilities.Fig. 5: *BWGossip* analysis in the presence of link failures.